# REPORT DOCUMENTATION PAGE

| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| 11-10-2012 | Journal Article | August 2012-October 2012 |

**4. TITLE AND SUBTITLE**

Higher Order Wide-Angle Split-Step Spectral Method

**5a. CONTRACT NUMBER**
FA8650-08-D-6930

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**
0602202F

**6. AUTHOR(S)**

B.H. Hokr, C.D. Clark, R.E. Grotheer and R.J. Thomas

**5d. PROJECT NUMBER**
7757

**5e. TASK NUMBER**
B2

**5f. WORK UNIT NUMBER**
37

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Research Laboratory
Human Effectiveness Directorate
Bioeffects Division
Optical Radiation Bioeffects              TASC
4141 Petroleum Rd                         4241 Woodcock Dr
JBSA Fort Sam Houston, TX 78234           San Antonio, Texas 78228

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFRL and TASC

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Air Force Research Laboratory
Human Effectiveness Directorate
Bioeffects Division
Optical Radiation Bioeffects
4141 Petroleum Rd
JBSA Fort Sam Houston, TX 78234

**10. SPONSOR'S/MONITOR'S ACRONYM(S)**

711 HPW/RHDO

**11. SPONSOR'S/MONITOR'S REPORT NUMBER(S)**

AFRL-RH-FS-JA-2012-0024

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Distribution A: Approved for public release; distribution unlimited (approval given by local Public Affairs Office PA# TSRL-PA-12-0068)

**14. ABSTRACT**
We develop a higher-order method for non-paraxial beam propagation based on the wide-angle split-step spectral (WASSS) method previously reported [Clark and Thomas, Opt. Quantum. Electron. 41, 849 (2010)]. The higher-order WASSS (HOWASSS) method approximates the Helmholtz equation by keeping terms up to third-order in the propagation step size, in the Magnus expansion. A symmetric exponential operator splitting technique is used to simplify the resulting exponential operators. The HOWASSS method is applied to the problem of waveguide propagation, where an analytical solution is known, to demonstrate the performance and accuracy of the method. The performance enhancement gained by implementing the HOWASSS method on a graphics processing unit (GPU) is demonstrated. When highly accurate results are required the HOWASSS method is shown to be substantially faster than the WASSS method.

**15. SUBJECT TERMS**

| 16. SECURITY CLASSIFICATION OF: Unclassified | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON Robert Thomas |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | U | 11 | **19b. TELEPHONE NUMBER** *(include area code)* |
| Unclassified | Unclassified | Unclassified | | | |

# Higher-order wide-angle split-step spectral method for non-paraxial beam propagation

**Brett H. Hokr,**[1,2,3*] **C. D. Clark III,**[3,4] **Rachel E. Grotheer,**[3,5] **and Robert J. Thomas** [6]

[1] *Department of Physics & Astronomy, Texas A&M University, College Station, TX, 77843, USA*
[2] *Department of Biomedical Engineering, Texas A&M University, College Station, TX, 77843, USA*
[3] *TASC Inc., San Antonio, TX, 78228, USA*
[4] *Department of Physics, Fort Hays State University, Hays, KS, 67601, USA*
[5] *Department of Mathematical Sciences, Clemson University, Clemson, SC, 29634, USA*
[6] *711 Human Performance Wing, Optical Radiation Bioeffects Branch, JBSA Fort Sam Houston, TX, 78234, USA*

*[*brett.hokr@tamu.edu](mailto:brett.hokr@tamu.edu)*

**Abstract:** We develop a higher-order method for non-paraxial beam propagation based on the wide-angle split-step spectral (WASSS) method previously reported [Clark and Thomas, Opt. Quantum. Electron., **41**, 849 (2010)]. The higher-order WASSS (HOWASSS) method approximates the Helmholtz equation by keeping terms up to third-order in the propagation step size, in the Magnus expansion. A symmetric exponential operator splitting technique is used to simplify the resulting exponential operators. The HOWASSS method is applied to the problem of waveguide propagation, where an analytical solution is known, to demonstrate the performance and accuracy of the method. The performance enhancement gained by implementing the HOWASSS method on a graphics processing unit (GPU) is demonstrated. When highly accurate results are required the HOWASSS method is shown to be substantially faster than the WASSS method.

© 2013 Optical Society of America

## References and links

1. G. R. Hadley, "Multistep method for wide-angle beam propagation," Opt. Lett. **17**, 1743–1745 (1992).
2. K. Q. Le, R. Godoy-Rubio, P. Bienstman, and G. R. Hadley, "The complex Jacobi iterative method for three-dimensional wide-angle beam propagation," Opt. Express **16**, 17021–17030 (2008).
3. Y. Y. Lu and P. L. Ho, "Beam propagation method using a $[(p-1)/p]$ Padé approximant of the propagator," Opt. Lett. **27**, 683–685 (2002).
4. A. Sharma and A. Agrawal, "New method for nonparaxial beam propagation," J. Opt. Soc. Am. B **21**, 1082–1087 (2004).
5. A. Sharma and A. Agrawal, "Non-paraxial split-step finite-difference method for beam propagation," Opt. Quantum. Electron. **38**, 19–34 (2006).
6. C. D. Clark and R. Thomas, "Wide-angle split-step spectral method for 2D or 3D beam propagation," Opt. Quantum. Electron. **41**, 849–857 (2010).
7. M. Guizar-Sicairos and J. C. Gutiérrez-Vega, "Computation of quasi-discrete Hankel transforms of integer order for propagating optical wave fields," J. Opt. Soc. Am. A **21**, 53–58 (2004).

8. W. Magnus, "On the exponential solution of differential equations for a linear operator," Comm. Pure Appl. Math. **7**, 649–673 (1954).
9. M. Bauer, R. Chetrite, K. Ebrahimi-Fard, and F. Patras, "Time-ordering and a generalized Magnus expansion," Lett. Math. Phys. **103**, 331–350 (2012).
10. M. J. Adams, *An Introduction to Optical Waveguides* (Wiley, 1981).

## 1. Introduction

Beam propagation methods (BPM) are a large class of numerical methods for solving the scalar Helmoltz equation, and are popular for simulating guided waves and laser beams, as they are typically both fast and efficient. Early methods made use of the paraxial approximation, which greatly simplifies the problem by reducing the propagation equation to first order. However, these methods were severely limited in their application. Any beam profile containing spatial frequencies with angles greater than a few degrees, with respect to the propagation axis, incur significant phase errors. Many methods have been developed to drop the paraxial approximation and include wide-angle waves. In several of these, the Helmholtz equation is formally rewritten as a first-order differential equation which includes the square root of an operator. The square root operator is either approximated using real or complex Padé approximants and a finite-difference or iterative method is used to solve the equation [1, 2], or the analytical solution is found which results in an exponential of the square root of an operator that is approximated with a Padé approximant [3]. Recently, Sharma extended an operator-splitting technique used on the paraxial wave equation to the non-paraxial wave equation (Helmholtz equation) [4]. The splitting allows diffraction and the refractive index variations to be handled separately. Various numerical methods can be used once the operator has been split, such as collocation or finite-difference [4, 5].

In a recent publication, we described a numerical beam propagation method that represents the beam profile in the basis of the eigenvectors of the Laplacian operator and uses a symmetric operator splitting technique to account for the refractive index variations, known as the wide-angle split-step spectroscopic (WASSS) method [6]. In general, the method provided a two-fold speedup to the finite-difference method reported by Sharma [4] This improvement could be increased by use of a fast Fourier transform algorithm. Here we develop a higher-order WASSS (HOWASSS) method that extends the approximation to higher-order, providing a more efficient method when high accuracies are required. We apply the HOWASSS method to the problem of waveguide propagation to demonstrate the performance and accuracy of the method. An additional performance enhancement is obtained by implementing the method on a graphics processing unit (GPU) using compute unified device architecture (CUDA) technology from NVIDIA™.

## 2. Formulation

Beam propagation in a medium with a non-uniform refractive index is described by the scalar Helmholtz equation,

$$\frac{\partial^2}{\partial z^2}\psi(z,\mathbf{r}) + \nabla_{\mathbf{r}}^2\psi(z,\mathbf{r}) + k_0^2\bar{n}^2\psi(z,\mathbf{r}) + k_0^2\left(n^2(z,\mathbf{r}) - \bar{n}^2\right)\psi(z,\mathbf{r}) = 0, \tag{1}$$

where $\psi(z,\mathbf{r})$ is the complex scalar electric field, $z$ is a Cartesian coordinate in the direction of propagation, $\mathbf{r}$ are the transverse coordinates, and $\nabla_{\mathbf{r}}^2$ is the transverse Laplace operator. As usual, $k_0$ is the free space wavenumber, $n(z,\mathbf{r})$ is the refractive index distribution, and $\bar{n}^2 = \min\left[n^2(z,x)\right]$. For simplicity we will only consider the two-dimensional (2D) case, however generalization to three dimensions (3D) follows quite trivially. In two dimensions, the electric field is denoted $\psi(z,x)$. Note that $x$ is not necessarily a Cartesian coordinate but could,

for example, be a radial coordinate. Expanding $\psi(z,x)$ in terms of the eigenfunctions of the transverse Laplace operator we write

$$\psi(z,x) = \sum_i a_i(z)\phi_i(x),\tag{2}$$

where

$$\nabla_x^2 \phi_i(x) = -\lambda_i^2 \phi_i(x)\tag{3}$$

$$\int \phi_i^*(x)\phi_j(x)\,\mathrm{d}x = \delta_{i,j}.\tag{4}$$

Multiplying the 2D form of Eq. (1) by $\phi_j^*(x)$, integrating over the transverse coordinate, and applying Eqs. (3) and (4) we are left with

$$\frac{\partial^2}{\partial z^2} a_j(z) = -\left(k_0^2 \bar{n}^2 - \lambda_j^2\right) a_j(z) - \sum_i a_i(z) \int k_0^2 \left(n^2(z,x) - \bar{n}^2\right) \phi_j^*(x)\phi_i(x)\,\mathrm{d}x.\tag{5}$$

To discretize the problem space, let $N_x$ and $N_z$ be the number of discrete grid points in x and z respectively, and let $i,j \in [1,N_x]$, and $l \in [1,N_z]$. We let $x_i = i\Delta x + x_0$ and $z_l = l\Delta z + z_0$, where $\Delta x$ and $\Delta z$ are the corresponding step sizes. The matrix $\mathbf{N}(z)$ is defined by

$$N_{i,j}(z) = k_0^2 \left(n^2(z,x_i) - \bar{n}^2\right) \delta_{i,j}.\tag{6}$$

The discrete eigentransform matrix, $\mathbf{S}$ [6], transforms a vector into the spectral basis, while $\mathbf{S}^{-1}$ transforms vectors back into the spatial basis. In the case of Cartesian coordinates with hard boundary conditions this matrix becomes the discrete Fourier transform matrix, and in cylindrical coordinates with azimuthal symmetry and hard boundary conditions, $\mathbf{S}$ takes the form of a discrete Hankel transform [7]. We also must define the constant matrix $\mathbf{M}$ with elements

$$M_{i,j} = \sqrt{k_0^2 \bar{n}^2 - \lambda_i^2}\,\delta_{i,j}.\tag{7}$$

Notice that if $\lambda_i > k_0 \bar{n}$ then $M_{i,j}$ becomes imaginary. If we allow eigenvalues such that $\lambda_i > k_0 \bar{n}$, then the trigonometric functions appearing in $\mathbf{P}$ will become hyperbolic (see Eq. (25)), making the method numerically unstable. For this reason, we will limit $N_x$ to satisfy $\lambda_i < k_0 \bar{n}$. However, we must include at least enough eigenfrequencies to sufficiently represent the initial conditions, otherwise large errors will be incurred. Using these definitions Eq. (5) becomes

$$\frac{\partial^2}{\partial z^2}\mathbf{a}(z) = -\mathbf{M}^2 \mathbf{a}(z) - \mathbf{S}\mathbf{N}(z)\mathbf{S}^{-1}\mathbf{a}(z).\tag{8}$$

Now if we define

$$\mathbf{A}(z) = \begin{bmatrix} \mathbf{a}(z) \\ \mathbf{M}^{-1}\frac{\partial}{\partial z}\mathbf{a}(z) \end{bmatrix}\tag{9}$$

$$\mathbf{H}(z) = \begin{bmatrix} \mathbf{0} & \mathbf{M} \\ -\mathbf{M} - \mathbf{M}^{-1}\mathbf{S}\mathbf{N}(z)\mathbf{S}^{-1} & \mathbf{0} \end{bmatrix},\tag{10}$$

we can write the Helmholtz equation as a first order vector differential equation

$$\frac{\partial}{\partial z}\mathbf{A}(z) = \mathbf{H}(z)\mathbf{A}(z).\tag{11}$$

The exact solution to Eq. (11) is given by a Magnus expansion [8],

$$\mathbf{A}(z) = \exp\left(\Omega_1(z, z_0) + \Omega_2(z, z_0) + \cdots\right)\mathbf{A}(z_0). \tag{12}$$

The first two terms of the expansion are

$$\Omega_1(z, z_0) = \int_{z_0}^{z} \mathbf{H}(t)\, dt \tag{13}$$

$$\Omega_2(z, z_0) = \frac{1}{2}\int_{z_0}^{z}\int_{z_0}^{t_1} [\mathbf{H}(t_1), \mathbf{H}(t_2)]\, dt_2\, dt_1. \tag{14}$$

Here $[\mathbf{H}(t_1), \mathbf{H}(t_2)]$ is the usual commutator. Note that Eq. (11) is essentially the time-dependent Schrödinger equation, in which context the Magnus expansion is more well-known as the time-ordered exponential operator [9]. However, we will refer to Eq. (12) as a Magnus expansion to be consistent with discussions concerning initial value problems of the form of Eq. (11).

$\mathbf{H}(z)$ can be written as the sum of two matrices, one that is constant and one that depends on $z$,

$$\mathbf{H}(z) = \mathbf{H}_1 + \mathbf{H}_2(z) = \begin{bmatrix} \mathbf{0} & \mathbf{M} \\ -\mathbf{M} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{S}\mathbf{N}(z)\mathbf{S}^{-1} & \mathbf{0} \end{bmatrix}. \tag{15}$$

Using this and the trapezoid rule to approximate the integral in Eq. (13) we obtain,

$$\Omega_1(z_{l+1}, z_l) = \mathbf{H}_1\Delta z + \left(\mathbf{H}_2(z_{l+1}) + \mathbf{H}_2(z_l)\right)\frac{\Delta z}{2} + \mathcal{O}\left((\Delta z)^3\right). \tag{16}$$

To approximate $\Omega_2(z_{l+1}, z_l)$ we first need to compute the commutator. After making use of Eq. (15) and simplifying we find

$$[\mathbf{H}(z_1), \mathbf{H}(z_2)] = \begin{bmatrix} \mathbf{S}(\mathbf{N}(z_1) - \mathbf{N}(z_2))\mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1}\mathbf{S}(\mathbf{N}(z_2) - \mathbf{N}(z_1))\mathbf{S}^{-1}\mathbf{M} \end{bmatrix}. \tag{17}$$

Note that $[\mathbf{H}_2(z_1), \mathbf{H}_2(z_2)] = 0$. We apply the trapezoid method twice to approximate the double integral in Eq. (14) and obtain

$$\Omega_2(z_{l+1}, z_l) = [\mathbf{H}(z_{l+1}), \mathbf{H}(z_l)]\frac{(\Delta z)^2}{8} + \mathcal{O}\left((\Delta z)^3\right). \tag{18}$$

The next term in the Magnus series will contribute a factor of $(\Delta z)^3$ as it involves a triple integral. Keeping terms up to $(\Delta z)^3$

$$\mathbf{A}(z_{l+1}) \approx \exp\left(\mathbf{H}_1\Delta z + \left(\mathbf{H}_2(z_{l+1}) + \mathbf{H}_2(z_l)\right)\frac{\Delta z}{2} + [\mathbf{H}(z_{l+1}), \mathbf{H}(z_l)]\frac{(\Delta z)^2}{8}\right)\mathbf{A}(z_l). \tag{19}$$

Because this contains the exponential of a dense matrix, which is difficult to handle numerically, we will split this exponential into a form that will be easier to work with via the symmetric exponential splitting technique

$$\exp\left((\mathbf{A} + \mathbf{B})\Delta z\right) = \exp\left(\mathbf{A}\frac{\Delta z}{2}\right)\exp\left(\mathbf{B}\Delta z\right)\exp\left(\mathbf{A}\frac{\Delta z}{2}\right) + \mathcal{O}\left((\Delta z)^3\right). \tag{20}$$

First, we split the $\Delta z$ terms in the exponential from the $(\Delta z)^2$ terms. Then we split the terms involving $\mathbf{H}_1$ from those involving $\mathbf{H}_2$, and we are left with

$$\mathbf{A}(z_{l+1}) \approx \mathbf{P}\mathbf{Q}(z_{l+1})\mathbf{Q}(z_l)\mathbf{P}\mathbf{C}(z_{l+1}, z_l)\mathbf{P}\mathbf{Q}(z_{l+1})\mathbf{Q}(z_l)\mathbf{P}\mathbf{A}(z_l), \tag{21}$$

where

$$P = \exp\left(H_1 \frac{\Delta z}{4}\right) \tag{22}$$

$$Q(z) = \exp\left(H_2(z) \frac{\Delta z}{4}\right) \tag{23}$$

$$C(z_1, z_2) = \exp\left([H(z_1), H(z_2)] \frac{(\Delta z)^2}{8}\right). \tag{24}$$

Each operator $P$, $Q(z)$, and $C(z_1, z_2)$ can be expressed by a $2 \times 2$ block matrix where the matrices within each operator are diagonal. This allows the exponential operators to be simply evaluated by expanding the exponential into its power series

$$P = \begin{bmatrix} \cos\left(M\frac{\Delta z}{4}\right) & \sin\left(M\frac{\Delta z}{4}\right) \\ -\sin\left(M\frac{\Delta z}{4}\right) & \cos\left(M\frac{\Delta z}{4}\right) \end{bmatrix} \tag{25}$$

$$Q(z) = \begin{bmatrix} I & 0 \\ -M^{-1}SN(z)S^{-1}\frac{\Delta z}{4} & I \end{bmatrix} \tag{26}$$

$$C(z_1, z_2) =$$
$$\begin{bmatrix} S\exp\left((N(z_1) - N(z_2))\frac{(\Delta z)^2}{8}\right)S^{-1} & 0 \\ 0 & M^{-1}S\exp\left((N(z_2) - N(z_1))\frac{(\Delta z)^2}{8}\right)S^{-1}M \end{bmatrix}. \tag{27}$$

Now all matrices inside of functions are diagonal making them quite simple to numerically evaluate. In this form the physical interpretation of these operators is most transparent. $P$ is the operator corresponding to a propagation of the pulse through a homogenous medium with an index of refraction of $\bar{n}$. $Q(z)$ takes into account the difference $n(z, x) - \bar{n}$, while $C(z_1, z_2)$ depends on the change in the refractive index over the step taken. Thus, in this sense we would like to draw the analogy that $Q(z)$ acts like the constant term of a Taylor's series expansion, and $C(z_1, z_2)$ acts in a manner similar to the first derivative term in such a series. We can save some additional matrix-vector multiplications by combining $Q(z_{l+1})Q(z_l)$

$$Q(z_{l+1})Q(z_l) = \begin{bmatrix} I & 0 \\ -M^{-1}S(N(z_{l+1}) + N(z_l))S^{-1}\frac{\Delta z}{4} & I \end{bmatrix}. \tag{28}$$

## 3. Numerical example

To test our method we will simulate beam propagation through a two-dimensional (Cartesian) symmetric Epstein-layer waveguide tilted at an angle $\theta$ from the positive $z$ axis [5]. Hard boundary conditions where $\psi(z, x_0) = 0$ and $\psi(z, x_f) = 0$ are assumed. The eigenfunctions of the transverse Laplace operator are then given by

$$\phi_i(x) = \sin\left(\frac{i\pi}{(x_f - x_0)}x\right). \tag{29}$$

The resulting eigentransform matrix is given by the discrete Fourier matrix

$$S_{i,j} = \sqrt{\frac{2}{N_x + 1}} \sin\left(\frac{\pi(i+1)(j+1)}{N_x + 1}\right) = S_{i,j}^{-1}. \tag{30}$$

Table 1. Parameters used in the numerical calculations

| $\bar{n} = 2.1455$ | $\Delta n = 0.003$ | $x_0 = 0\,\mu\text{m}$ | $x_f = 300\,\mu\text{m}$ |
|---|---|---|---|
| $w = 5\,\mu\text{m}$ | $k_0 = 4.88128\,\mu\text{m}^{-1}$ | $z_0 = 0\,\mu\text{m}$ | $z_f = 100\,\mu\text{m}$ |

This particular Fourier matrix was chosen because the hard boundary conditions at $x_0$ and $x_f$ are automatically satisfied. We do not need to include these points in our grid, defined by $\Delta x = \left(x_f - x_0\right)/\left(N_x + 1\right)$ and $\Delta z = \left(z_f - z_0\right)/N_z$. The refractive index profile is

$$n(z,x) = \sqrt{\bar{n}^2 + 2\bar{n}\left(\Delta n\right)\operatorname{sech}^2\left(\frac{2\left(\tilde{x}\cos\left(\theta\right) - z\sin\left(\theta\right)\right)}{w}\right)}, \tag{31}$$

where $\tilde{x}$ is a shifted coordinate to align the waveguide in the center of our computational region to make the hard boundary conditions as negligible as possible, $\Delta n$ is the height of the refractive index shift, $w$ is the width of the waveguide. The shifted coordinate is given by $\tilde{x} = x - (1/2)\left(x_f - x_0\right) + (1/2)\left(z_f - z_0\right)\tan\left(\theta\right)$. The initial electric field is given by the zeroth order mode of the Epstein-layer waveguide

$$\psi(0,x) = \operatorname{sech}^W\left(\frac{2\tilde{x}\cos\left(\theta\right)}{w}\right)\exp\left(iK_0\tilde{x}\sin\left(\theta\right)\right), \tag{32}$$

Here, $i$ is the unit imaginary number, not to be confused with the counting index used elsewhere. $W$ and $K_0$ are given by

$$W = \frac{1}{2}\left(\sqrt{1 + 2w^2k_0^2\bar{n}\Delta n} - 1\right) \tag{33}$$

$$K_0 = \sqrt{\left(\frac{2W}{w}\right)^2 + \left(k_0\bar{n}\right)^2}. \tag{34}$$

The exact solution for this tilted Epstein-layer waveguide is [10]

$$\psi_e(z,x) = \operatorname{sech}^W\left(\frac{2\left(\tilde{x}\cos\left(\theta\right) - z\sin\left(\theta\right)\right)}{w}\right)\exp\left(iK_0\left(\tilde{x}\sin\left(\theta\right) + z\cos\left(\theta\right)\right)\right). \tag{35}$$

To measure the error we use the correlation factor,

$$\text{Error}(z) = \left|1 - \frac{\left(\int_{x_0}^{x_f}\psi^*(z,x)\,\psi(z,x)\,dx\right)^2}{\left(\int_{x_0}^{x_f}\psi_e^*(z,x)\,\psi_e(z,x)\,dx\right)^2}\right|, \tag{36}$$

as this provides a measure for both the profile shape and amplitude of the beam [5, 6].

C and CUDA versions of both the WASSS and HOWASSS methods were implemented. The code was compiled using nvcc version 4.2 for CUDA code and gcc version 4.6.3 for the C code. The code was run on a workstation equipped with an Intel Xeon X5960 CPU, which is a hyper-threaded six-core CPU clocked at 3.47 GHz with 48 GB of RAM, and a NVIDIA QUADRO 6000 GPU, which has 448 CUDA cores at 574 MHz core clock speed (750 MHz memory clock speed) and 6 GB of dedicated GPU DDR5 RAM.

The simulation was run using the parameters listed in Table 1. The value of $k_0$ was selected to ensure that we could include 1000 transverse modes while still keeping **M** real-valued. When
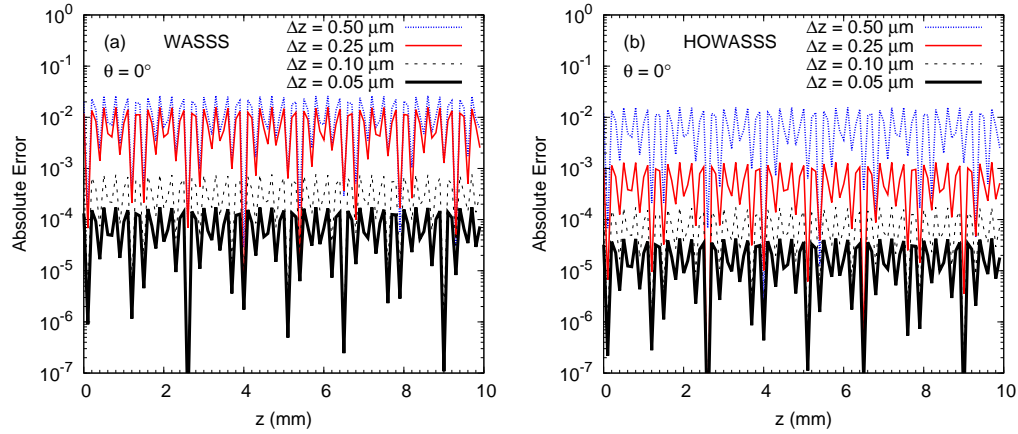
Fig. 1. Error as a function of propagation distance for an aligned waveguide with $N_x = 1000$ for the (a) WASSS and (b) HOWASSS methods. Note that we have sampled the error every 0.5 mm, and not at every z step calculated, because the rapid oscillations would make the graph difficult to read otherwise.

the waveguide was aligned ($\theta = 0°$) roughly an order of magnitude decrease in the error was observed with the HOWASSS method over the WASSS method (see Fig. 1). However, the improvement was more pronounced when we tilted the waveguide at an angle of 50 degrees, especially for large step sizes (see Fig. 2). Figure 3 shows the calculated electric field next to the exact solution to further illustrate the accuracy of the HOWASSS method. The simulations shown in Figs. 1, 2, and 3 were run using double precision accuracy numbers on the GPU. Both methods were capable of producing accurate results using single precision arithmetic, until about $\Delta z = 0.05\,\mu m$. At this point, round-off error began to affect the results of the HOWASSS method due to the increased number of matrix-vector multiplications required in each time step. The WASSS method does not suffer as quickly from this problem because it is computationally more simple. The limit where round-off error begins to affect the double precision code was not observed for either method.
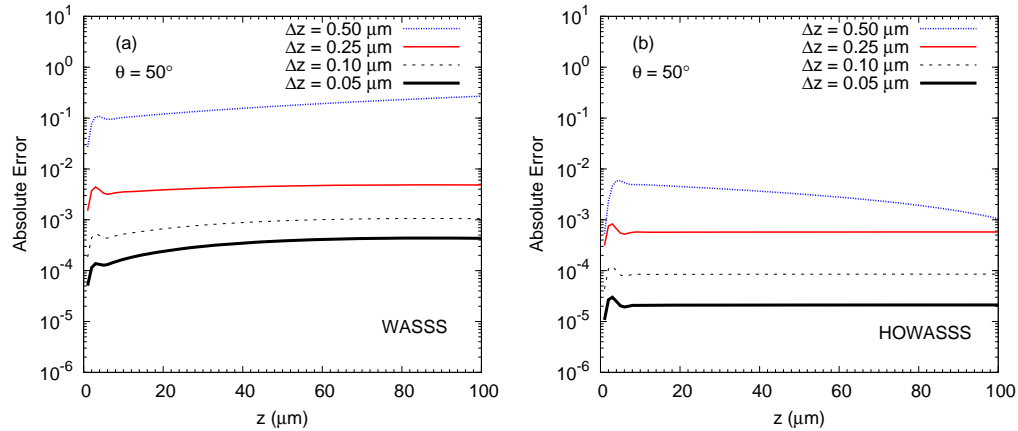


Fig. 2. Error as a function of propagation distance for a waveguide rotated 50 degrees with $N_x = 1000$ for the (a) WASSS and (b) HOWASSS methods.
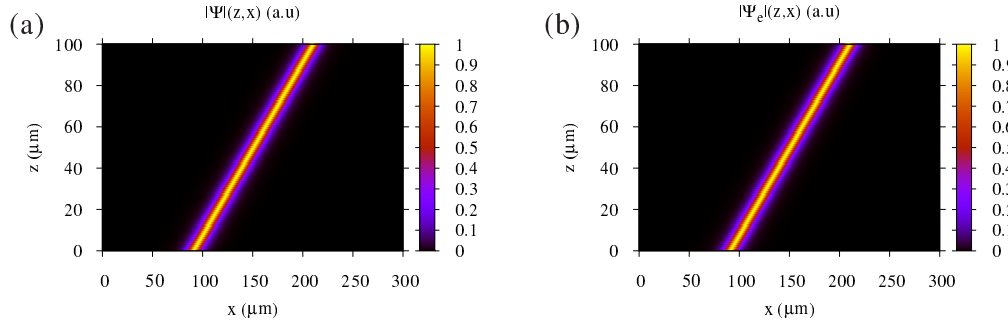
Fig. 3. Plot of $|\psi(z,x)|$ with $N_x = 1000$, $N_z = 2000$, and $\theta = 50°$ for the (a) HOWASSS and (b) exact solutions for a waveguide tilted at 50 degrees.

## 4. Discussion

### 4.1. Stability

By virtue of being a higher order method, the HOWASSS method has improved stability over the WASSS method (see Fig. 4). As we increase the tilt of the waveguide, the WASSS method performs more poorly as the angle increases, and at large step sizes even shows signs of being a bit unstable. However, the HOWASSS method actually becomes more accurate at larger angles. Traditionally, beam propagation methods struggle to obtain accurate results when there are rapid changes in the index of refraction. To simulate a rapidly changing index of refraction we increase the change in the index between the waveguide and the surrounding medium, $\Delta n$, while keeping the width of the waveguide fixed. We see that at a $50°$ waveguide tilt the WASSS method actually becomes unstable with a large stepsize, while the HOWASSS is able to remain stable for at least an additional order of magnitude increase in $\Delta n$. However, both methods do loose accuracy as the change in the refractive index becomes steeper, but by decreasing $\Delta z$ accurate results can still be obtained in a reasonable run time. This same analysis was done for a $0°$ tilted waveguide and the results were quite similar and so are not shown here.
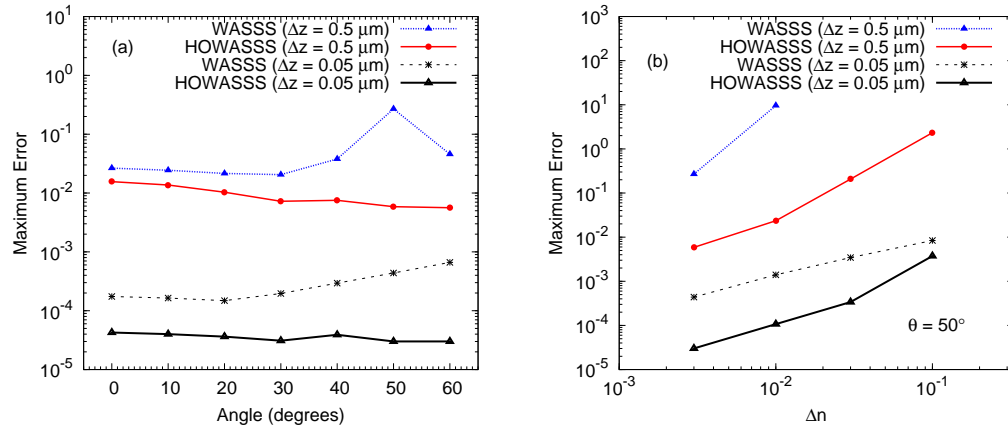


Fig. 4. (a) The maximum error obtained as a function of waveguide tilt angle showing that the HOWASSS method actually gains a small amount of accuracy at larger angles. (b) The maximum error obtained as a function of waveguide depth, $\Delta n$.
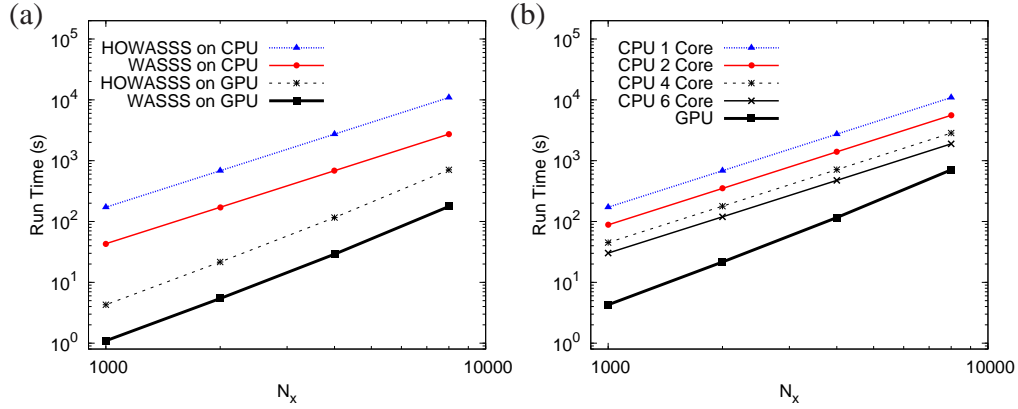
Fig. 5. (a) Run times of HOWASSS and WASSS methods on the GPU and the single-core CPU for 1000 propagation steps. (b) Comparison of HOWASSS method run times using different number of cores on the CPU and using the GPU for 1000 propagation steps.

## 4.2. Speed

Computationally, the HOWASSS method requires the multiplication of matrices with vectors. If we restrict $N_x$ so that the matrix $\mathbf{M}$ is real then, for the example in Section 3, all the matrices will be real, however the vector $\mathbf{A}$ will be complex. In general, the eigenfunctions and the refractive index could be complex. However, for simplicity we will assume these to be real.

Diagonal matrix multiplication is equivalent to element-by-element vector multiplication. Hence, multiplying an $N_x \times N_x$ diagonal matrix by an $N_x$ element complex vector requires $2N_x$ operations. Multiplying an $N_x \times N_x$ dense matrix by an $N_x$ element complex vector is equivalent to performing $2N_x$ dot products, each requiring $N_x$ multiplications and $N_x - 1$ additions, for a total of $2N_x(2N_x - 1) = 4N_x^2 - 2N_x$ operations. Applying $\mathbf{P}$ requires a total of 4 diagonal matrix-vector multiplications for a total of $8N_x$ operations. Applying $\mathbf{Q}(z_{l+1})\mathbf{Q}(z_l)$ requires 2 diagonal matrix-vector multiplications, 2 dense matrix-vector multiplications, 2 vector-vector additions, and 1 scalar-vector multiplication for a total of $8N_x^2 + 6N_x = 2N_x(4N_x - 3)$ operations. Applying $\mathbf{C}(z_{l+1}, z_l)$ requires 4 dense matrix-vector multiplications, 1 vector-vector addition, and 2 element-by-element exponential operations (assumed to only be 1 operation per element) for a total of $16N_x^2 - 2N_x = 2N_x(8N_x - 1)$ operations. $\mathbf{P}$ is applied four times, $\mathbf{Q}(z_{l+1})\mathbf{Q}(z_l)$ is applied twice, and $\mathbf{C}(z_{l+1}, z_l)$ is applied once each step, giving a total of $32N_x^2 + 42N_x = 2N_x(16N_x + 21)$ operations. Following the same logic for the WASSS method, we find that it requires $8N_x^2 + 6N_x = 2N_x(4N_x + 3)$ operations. Note that this operation count differs from the one reported by Clark and Thomas [6] because we are counting both multiplications and additions in this calculation.

Furthermore, to leading order in $N_x$, the HOWASSS method is approximately a factor of 4 slower for a given $\Delta z$ (see Fig. 5). However, the initial hypothesis was that a significant improvement in accuracy may lead to a more efficient algorithm. To test this hypothesis, we executed the HOWASSS and WASSS methods using the GPU implementation with various step sizes, holding all other aspects of the problem constant. Figure 6 summarizes the result. For a propagation length of 100 micrometers, and a waveguide tilt angle of 50 degrees, we show the compute time per micron of propagation as a function of the maximum absolute error. For two different values of $N_x$, the efficiency at constant error is better for the HOWASSS method, as indicated by shorter compute times. In fact, for error values smaller than about $10^{-4}$, the HOWASSS method is much better, with efficiency rapidly exceeding an order of magnitude for
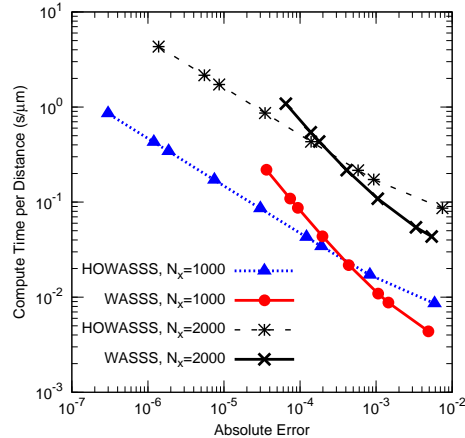
Fig. 6. Comparison of the HOWASSS method and WASSS method compute time per propagation distance with $N_x = 1000$ and $N_x = 2000$. Both methods were run on the GPU.

absolute error values of less than $10^{-6}$.

The data presented in this paper was generated using double precision arithmetic. This does not result in a significant difference in the run time when the code is run on a CPU. However, GPUs are intrinsically designed to deal with single-precision arithmetic. In order to compute one double-precision number the GPU must perform two single-precision calculations and some additional overhead to combine the result, leading to, at a minimum, a factor of two speed-up when single-precision numbers are used. This speed-up also depends on the specific graphics card used, as some have less capability than others when it comes to double precision arithmetic. Rounding errors can affect the HOWASSS method if $N_z$ is large, around 2000 for the specific example considered in this paper. If the application does not require extreme accuracy, then a substantial speed-up can be achieved by using single-precision arithmetic on the GPU.

### 4.3. Parallelization

Both the WASSS and HOWASSS methods readily lend themselves to parallelization. For our implementation we chose to use both OpenMP, to make use of multi core CPUs, and NVIDIA compute unified device architecture (CUDA) to make use of the processing power of the graphics processing unit (GPU). Modern GPUs possess orders of magnitude more computational power than the typical CPU. However, to completely utilize this power the algorithm must possess a very high level of parallelism to completely saturate the many processing units of the GPU.

Unfortunately, neither the WASSS nor the HOWASSS method are ideal for utilizing the full power of the GPU because many of the matrix-vector multiplications involve diagonal matrices. Faster than their dense counterparts, these computationally reduce to simple element-by-element vector multiplication. While this is a perfectly parallel operation, it does not offer the large number of independent calculations needed to saturate the GPU. These element-by-element vector multiplications suffer additionally from the fact that they require two memory reads and one memory write to compute only one multiplication. On the GPU reading and writing to memory is much slower than math operations. Given this, there is still a significant speed-up when the code is run on the GPU versus on a single core of the CPU (see Fig. 5). This speed-up was accomplished with little attention payed to optimization of the code. With further optimization an additional factor of two or more would be likely. Additionally, extending this

technique to even higher-orders might yield additional improvements.

### 4.4. *Generalizations to other coordinate systems, boundary conditions, and higher dimensions*

Both the WASSS and HOWASSS methods can be generalized to different coordinate systems or different boundary conditions. In fact the only complication is that the eigenfunctions and eigenvalues must be known. For a more detailed discussion on this topic see Clark and Thomas [6].

## 5. Conclusion

We have presented a method that is more efficient than previous non-paraxial beam propagation methods. The method casts the analytic solution of the Helmholtz equation as a Magnus expansion. Keeping terms up to $(\Delta z)^3$ in the Magnus expansion we use a symmetric operator splitting technique in order to analytically reduce the exponential matrices into a more simple form. The solution to the Helmholtz equation is then approximated via straightforward matrix multiplication. We have demonstrated the method in a simple geometry with 2D Cartesian coordinates with hard boundary conditions. The results obtained show our higher-order approach significantly improves the overall efficiency, when measured as compute time per distance of propagation, in cases where high accuracy results are required. The method can be easily extended to more generalized coordinate systems, higher dimensions, and various boundary conditions, provided that the eigenfunctions and eigenvalues of the transverse Laplace operator can be found for that geometry.

### Acknowledgments